

REMARKS/ARGUMENTS

1. SUMMARY

In the Final Office Action mailed September 23, 2004, the Examiner rejected claims 1, 2, 4 - 11, 16 - 20, and 22 - 35 under 35 U.S.C § 102(b) as being anticipated by U.S. Patent No. 6,073,163 to Clark et al. (hereinafter, "Clark"). In addition, the Examiner rejected claims 12 - 15 and 21 under 35 U.S.C. § 103(a) as being obvious in view of various combinations of Clark and U.S. Patent No.'s: 6,708,161; 6,715,145; 6,609,158; 6,535,913; 6,609,158; and 6,535,913.

In this reply, claims 5, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30 and 34 have been amended. Claim 35 has been canceled. Claims 1, 2, and 4 - 34 remain pending in the application.

2. RESPONSE TO § 102 REJECTIONS

To anticipate a claim, the reference must teach every element of the claim. "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

Claim 1 is not anticipated by Clark, because Clark does not disclose each and every limitation of claim 1.

Applicants' invention, as claimed, relates to a messaging system for updating user interface elements, for example, displayed in a web page of a web browser. Accordingly, in one embodiment of the invention, to update various interface elements, an interface update message may be communicated from one client computer system, via an intermediary computer system, to several client computer systems executing client application programs. As described in greater detail in paragraph 78 on page 28 of Applicants' specification, an interface

update message may originate at one client computer system, and then be communicated to one or more different client computer systems.

Claim 1 states:

1. A method to facilitate the update of a plurality of user interface categories utilizing a single client application program, the method including:
 - at a first computer system, generating a user interface data message wherein the user interface data message includes the plurality of user interface categories and the single client application program, wherein the single client application program executes at a second computer system, and wherein each user interface category includes a user interface element;
 - communicating the user interface data message from the first computer system to the second computer system;
 - receiving a user interface update message, at the first computer system, wherein the user interface update message includes an update to the plurality of user interface categories; and
 - communicating the user interface update message from the first computer system to the second computer system to enable the single client application program, at the second computer system, to update the plurality of user interface categories, by communicating the update to a script and the script, in turn, updating the respective user interface elements.

(Claim 1, emphasis added). According to claim 1, an interface update message is received at a first computer system, and then, the interface update message is communicated from the first computer system to the second computer system. For example, the same message that is received at the first computer system is communicated to the second computer system. Figure A, illustrated below, shows a simple block diagram corresponding to claim 1.

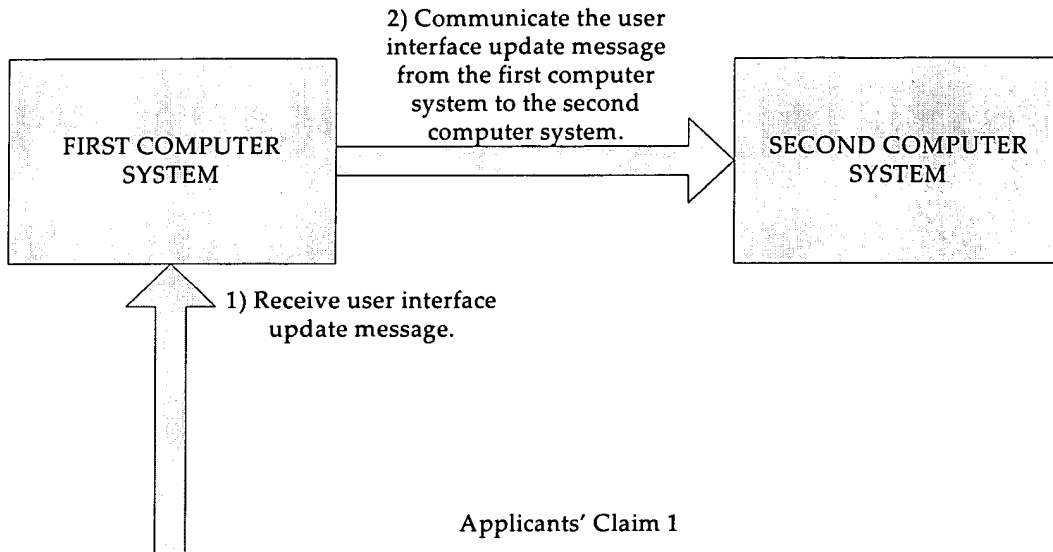


FIGURE A

Clark, on the other hand, discloses a system for updating user interface elements utilizing requests and responses. Specifically, Clark states:

During execution, client-side code * and server-side code * communicate with each other in a series of **requests and responses**. Specifically, requests are transmitted from client-side code * to server-side code * in the form of **UI event messages**, or requests for specific data. Responses from the server-side code * typically take the form of **UI change messages** which may specify changes to displayed values or may add or remove user interface components from the display.

(Clark, Col. 7, Lines 6 - 15, emphasis added). Figure B, set out below, illustrates a system consistent with Clark.

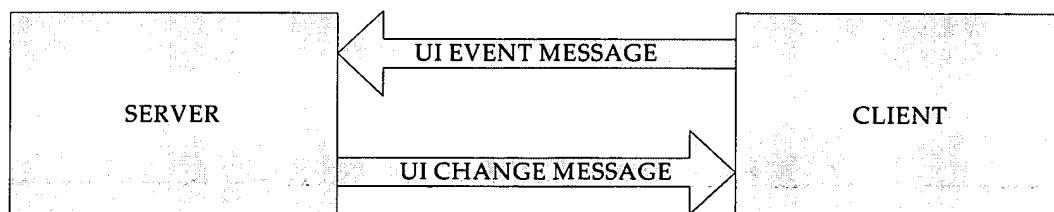


FIGURE B

According to Clark, first, a UI event message is communicated from a client to a server. For example, a (server) computer receives a UI event message. Next, UI logic on the server determines an appropriate response to the UI event message, and the server communicates a second message, for example, a UI change message, back to the client. Therefore, a second (client) computer receives a UI change message. According to Clark, the UI event message and the UI change message are two distinct and separate messages.

However, Applicants' claim 1 involves only an interface update message. Furthermore, according to claim 1, the interface update message is received at a first computer system, and then communicated to a second computer system. In contrast to claim 1, Clark does not disclose receiving a user interface update message, at a first computer system, and communicating the user interface update message from the first computer system to a second computer system to enable a single client application program, at the second computer system, to update a plurality of user interface categories. Consequently, Clark does not anticipate claim 1. Because claims 2 - 19 depend directly or indirectly on claim 1, Clark does not anticipate claims 2 - 19.

Claims 20, 22 - 25, and 30 are not anticipated by Clark, because Clark does not disclose each and every limitation of claims 20, 22 - 25, and 30.

Claim 20 states:

20. A method to facilitate updating a plurality of user interface categories utilizing a single client application program, the method including:
 at a first computer system, receiving a user interface data message from second computer system, wherein the user interface data message includes the plurality of user interface categories and the single client application program, wherein the single client application program executes at the first computer system, wherein each user interface category includes a user interface element;

sending a subscription subject message from the first computer system to the second computer system, the subscription subject message including a list of subscription subjects subscribed to by each user interface element of each category;

at the first computer system, a single client application program receiving a user interface update message, for subjects that have been subscribed to, from the second computer system; and

the single client application program updating the plurality of user interface categories, by communicating the update to a script and the script, in turn, communicating the update to the plurality of user interface categories to update the respective user interface elements.

(Claim 20, as amended, emphasis added).

Claim 20 involves sending a subscription subject message from a first computer system to a second computer system, wherein the subscription subject message includes a list of subscription subjects subscribed to by each user interface element of each category. For example, in one embodiment of Applicants' invention, a script executing on a client computer creates a subscription subject message including a list of subscription subjects to which each user interface element subscribes. Accordingly, a client application program, executing on a client computer, may send the subscription subject message to an agent process executing on another computer (e.g., a server computer). Consequently, the server computer, which receives the subscription subject message, is able to direct or forward a user interface update message to a first client computer system, even when the user interface update message originated on a second client computer system. Additionally, a user interface update message originating on a first client computer system may be multicast to a variety of client computer systems, based on subscription subjects. (See Applicants' Specification, Page 24, Paragraphs 60 - 62).

In contrast, Clark discloses a system that uses requests and responses (e.g., UI event messages and UI change messages) between one client computer and one server computer to update user interface elements on the client computer. According to Clark, user interface elements on a client computer are updated or changed only in response to a request sent by the client computer. In particular, Clark states:

The data initially required by dispatcher 514 may include, for example, code associated with particular user interface classes, such as buttons, as well as image data required to display those user interface classes.

The JAVA interpreter within browser 510 requests the specified information from web server 506. The web server 506 responds by sending the requested information to web browser 510. Web browser 510 supplies the information to the JAVA Applet 512, and Dispatcher 514 invokes the necessary code.

When the user interacts with the user interface generated by Applet 512, the Applet 512 generates a UI event message, as described above. The UI event message is transmitted to an application server 508. ...

...

The Forms Instance determines how to respond to the user input specified in the UI event message. While processing the UI event message, the Forms Instance generates UI definition records, as described above. The set of UI definition records generated in response to the user input is then sent back to dispatcher 514 in a UI change message. Dispatcher 514 dispatches the UI definition records within the UI change message to the appropriate UI objects. If any of the UI definition records require a UI class that has not yet been loaded onto client 504, the dispatcher 514 causes the web browser 510 to request the necessary object class code from the web server 506. The UI object code that resides within Applet 512 then processes the UI definition records by performing the UI-related operations specified therein.

(Clark, Col. 10, Lines 19 – 67).

Clark does not disclose or suggest sending a subscription subject message to a server computer. Therefore, Clark does not anticipate claim 20. For the same reasons, Clark does not anticipate independent claims 22, 23, 24, 25 and 30. Because claims 21 and 31 - 33 depend on claim 20 and 30 respectively, Clark does not anticipate claims 21 and 31 - 33.

Claims 26 - 30 and 34 are not anticipated by Clark, because Clark does not disclose each and every limitation of claims 26 - 29.

Claim 26 states:

26. A system to facilitate reception of an update from a plurality of user interface categories, the system comprising:

an agent process, at a first computer system, to communicate a user interface data message to a second computer system wherein the user interface data message includes a first client application program and a plurality of user interface categories, wherein each user interface category includes a user interface element; and

the agent process, in turn, to receive a user interface update message from the first client application program wherein the user interface update message is identified with a subject and generated, at the second computer system, by a user interface element adapted to communicate an update to the first client application program, wherein the agent process is to communicate the user interface update message to a communication process at the first computer system, and the communication process is to multicast the user interface update message to any computer system executing a client application program that has subscribed, with the first computer system, to receive user interface update messages identified with the subject.

(claim 26, as amended, emphasis added). According to claim 26, a communication process multicasts a user interface update message to any computer system executing a client application program that has subscribed, with a first computer system, to receive user interface update messages identified with a subject.

Clark, in contrast to claim 26, does not disclose or suggest multicasting a user interface update message or subscribing for messages. According to Clark, the "client-side code * establishes a permanent TCP/IP sockets communication with the server-side code * after the client-side code * is initiated." (Clark, Col. 11, Lines 51 - 53). Alternatively, Clark suggests using HTTP to communicate between a client and a server. In either case, Clark suggests using point-to-point communications only. Clark does not disclose or suggest multicasting, in part because Clark does not disclose or involve a system that multicasts user interface update messages to multiple computers based on subscription subjects. In particular, Clark does not disclose a communication process multicasting a user interface update message to any computer system executing a client application program that has subscribed, with a first computer system, to receive user interface update messages identified with a subject, as is recited in claim 26. Consequently, Clark does not anticipate claim 26. For the same reason, Clark does not anticipate independent claims 27 - 30 and 34. Because claims 31 - 33 depend on claim 30, Clark does not anticipate claims 31 - 33.

For the reasons provided above, Applicants submit that Clark does not disclose or suggest each and every limitation of claims 1, 2 and 4 - 34. Therefore, Clark does not anticipate the claims under 35 U.S.C. § 102. Accordingly, Applicants submit claims 1, 2 and 4 - 34 are in a condition for allowance, which is respectfully solicited.

3. RESPONSE TO § 103 REJECTIONS

To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The